

**IN THE SPECIFICATION:**

Please amend the specification as shown hereinbelow. Please note that some of the specification amendments presented hereinbelow further amend portions of the specification that were previously amended by Applicants in Applicants' Response, dated May 13, 2008, to the Office Action dated February 28, 2008.

Please delete the Summary of the Invention section, which was first added to the specification (after the Field of the Invention section and before the Brief Description of the Drawings section) in Applicants' Response, dated May 13, 2008, to the Office Action dated February 28, 2008.

Please amend the Brief Description of the Drawings section of the specification as follows (note that this amendment amends the version of the Brief Description of the Drawings section of the specification provided by Applicants in Applicants' Response, dated May 13, 2008, to the Office Action dated February 28, 2008):

**BRIEF DESCRIPTION OF THE DRAWINGS**

FIG. 1 depicts a block diagram of an information distribution system according to an embodiment of the present invention;

FIG. 2 depicts a block diagram of a receiver suitable for use in the information distribution system of FIG. 1;

FIG. 3 depicts a block diagram of an information sending system suitable for use in the information distribution system of FIG. 1;

FIG. 4 depicts a flow diagram of a data identifier (DID) and packet identifier (PID) processing method according to an embodiment of the invention;

FIG. 5 depicts a flow diagram of a packet receiving method according to an embodiment of the invention;

FIG. 6 is a block diagram of sample data being transmitted via several PIDs; and

FIG. 7 is a block diagram of a DC2 packet structure suitable for use with the present invention;

FIG. 8 depicts a functional block diagram of a transform according to an embodiment of the present invention; ~~and~~

FIG. 9 depicts functional block diagram of a transform according to an embodiment of the present ~~invention~~ invention; and

FIG. 10 depicts a block diagram of a MPEG2 packet suitable for use in one embodiment of the present invention.

Please amend the paragraph beginning on Pg. 4, Line 11 of the specification as follows:

DATA CAROUSEL. In one embodiment of the invention, local server 46 includes a carousel delivery application for delivering a carousel 50 (illustrated in ~~FIG. 2)~~ FIG. 3) of data files from local file server 46 to set-top box 28. Data from files residing on local server 46 are multiplexed onto at least one channel by multiplexer 57 to produce a carousel data stream for modulation onto a six megahertz channel of the transmission medium 24. The term "carousel" refers to a system in which a data set is repeatedly broadcast so that it may be accessed as required by many receivers. In that manner, files are sequentially relayed in cyclical fashion and thereby made available to set top box 28. A subscriber 26 can interact with the set-top box 28 to selectively display file information from the carousel on the screen of the television set 30. A variety of conventional carousel techniques exist. One example mechanism for periodically broadcasting the data via the broadcast delivery network is defined in the MPEG-2 DSM-CC specification (ISC/IEC 13818-6 IS). Other hardware and software mechanisms for providing a data carousel are suitable for use in embodiments of the invention employing a carousel.

Please amend the paragraph beginning on Pg. 5, Line 24 of the specification as follows:

Although not necessary to the invention, it is significant that with the DCT-2000 set-top box the hardware is unmodified, so that the existing base of publicly distributed set-top boxes may implement the invention without requiring upgrade servicing or replacement. However, to provide enhanced services to subscribers 26, the operation of the box 28 is typically modified by additional application

software downloaded thereto. In one embodiment, the application software provided to set top box 28 includes an ~~electronic program guide~~ Electronic Program Guide (EPG) application program 62 which communicates with an operating system 64 of the box 28 by placing calls through an application programming interface (API) 66. An electronic program guide (EPG) is an application that provides an on-screen listing of all programming and content that an interactive television service subscriber or digital television viewer has available to them. In some ~~cases~~ cases, an EPG includes visual images relating to the promotion, listing or selection of television programs or services, or other services where more than one service is available.

Please amend Pg. 6, Line 25 – Pg. 7, Line 22 of the specification as follows:

A microprocessor 74 controls the tuning operation of the tuner 70 and the OOB tuner (not shown) based on commands received from a subscriber via an input device such as a keypad or an infrared remote control device 76, as described below. To receive commands from a subscriber, the set-top box 28 includes an infrared sensor 78 connected to an infrared receiver 80 that provides the command signaling information to the microprocessor 74. In one embodiment of the invention, memory system 82 includes the VRTX operating system 64 stored therein, and generally comprises a combination of volatile dynamic RAM 84 and non-volatile RAM (NVRAM) 86. As those of ordinary skill in the art will readily ~~appreciated~~, appreciate, various types and configurations of memory are commercially available and suitable for use in memory system 82.

In accordance with digital broadcasts wherein digitized channels are multiplexed as data packets onto a six megahertz analog channel, the set-top box 28 also includes at least three packet identification (PID) filters 88-90 to extract the appropriate encoded data packets for a user-selected digital channel. Based on the user-selected display, audio and other requirements, microprocessor 74 writes an a packet identification value (PID) to each of the PID filters 88-90, whereby the PID filters 88-90 pass only those packets corresponding to the written PID value. As shown in ~~FIG. 3~~, FIG. 2, one of the

PID filters, filter 88, provides the filtered packets to an audio decoder 92 which decodes the digital audio data while another PID filter 90 provides filtered packets (for example, MPEG2 encoded packets) to the video decoder ~~[[52]]~~ 93.

At least a third PID filter 89 is provided to extract other data, including data from files stored on file server 46. A packet processor 94 handles those packets. The set-top box is also equipped with an on-screen display frame buffer (OSD) 96 capable of superimposing alphanumeric characters, other symbols and bitmap graphics over a displayed image. To accomplish this superimposition, an overlay 98 is provided to appropriately combine the video outputs of the video decoder ~~[[52]]~~ 93 and the OSD 96.

Please amend Pg.7, Line 23 – Pg. 8, Line 13 of the specification as follows:

GENERAL FUNCTION. The cable box 28 functions when the user provides an appropriate command to the cable box 28. For example, in response to a digital channel selection command, the microprocessor 74, tunes the in-band tuner 70 to an appropriate analog channel based on the digital channel selected by the subscriber 26. If a digital channel was selected, a table or the like stored in the memory 82 determines the analog channel that carries the digital channel's packets, along with the packet identification numbers (PIDs) corresponding to the selected digital channel, for writing into the PID filters 88 and 90. Once the PIDs have been written, the audio and video decoders ~~52 and~~ 92 and 93 will receive the appropriate packets and decode and output appropriate signals.

~~"ELECTRONIC PROGRAM GUIDES"~~ ELECTRONIC PROGRAM GUIDES. Advances in technology continue to create a wide variety of services and programs offered to users via television and other video equipment, such as video recorders connected to set top box 28. Such content is disseminated via cable, satellite, broadcast, and terrestrial systems. Content includes, but is not limited to, traditional broadcast and cable television programs, video services such as pay per view (PPV), near video on demand, promotional channels, game channels, localized or specially formatted information, ~~cable-delivered~~ cable-

delivered Personal Computer (PC) based content and services, Internet based content and services, and interactive services. Electronic Program Guides (EPGs) provide users with a means for viewing, selecting, interacting with and otherwise controlling available programs, features and services available to them.

Please amend the paragraph beginning on Pg. 8, Line 28 of the specification as follows:

ASSOCIATING PIDS WITH ~~FILENAMES~~ FILENAMES. Regardless of the application running in set top box 28, in order to obtain the contents of a desired file, the PID corresponding to the data stream carrying the desired file in the carousel must be written to PID filter 89. According to one technique, a PID associated with a desired file is obtained from a cross reference table 61 or the like stored in the memory 82. The cross reference table contains directory information that cross references a file identifier, such as a filename, to a corresponding PID that carries the desired file's packets. Typically, such a cross reference table is transmitted from headend 22 to set top box 28 on an Out of Band (OOB) communication link. However, in some embodiments of the invention, the cross reference table is transmitted on an in-band channel.

Please amend Pg. 9, Line 18 – Pg. 10, Line 22 of the specification as follows:

TRANSMIT END. FIG. 3 ~~illustrates in~~ illustrates in more detail, an embodiment of sender system 22 illustrated in FIG. 1. FIG. 3 illustrates a system and method for associating a filename with a PID, without the need for an OOB communications link. As previously described, local file server 46 includes at least one storage medium 53, file manager 49, and in some embodiments, at least one authoring system ~~[[65]]~~ 60. Files comprising scripts are stored on storage medium 53. Script files are downloaded from an external script source 47, source, for example information source 47 and ~~national~~ regional server 48. In some embodiments, files are provided to storage medium 53 by authoring system ~~[[65]]~~ 60. File manager 49 provides files from storage medium 53 to the remainder of system 22 for transport to subscriber set top boxes 28.

~~TRANSFORM.~~ TRANSFORM. File contents 65, e.g., data, of respective files from storage medium 53 are provided to packetizer 52. A file identifier, e.g., filename corresponding to each respective file to be transported is provided to transform 54. In one embodiment of the invention filenames are provided as American Standard Code for Information Interchange (ASCII) coded bits. As those of ordinary skill in the art will recognize, other types of bit coding techniques exist and are appropriate for use in the present invention to represent filenames. Transform 54 operates on a filename at its input to provide, at its output a plurality of bits representing a PID 55. In one embodiment of the invention, PID 55 comprises 13 bits in accordance with the MPEG2 systems specification. In addition to PID 55, alternative embodiments of the invention include bits representing a ~~Data Identifier (DID)~~ Payload Identifier (PIF) 56 and a Multicast Identifier (MCI) 57. ~~DID~~ PIF 56 and MCI 57 are utilized to further distinguish filenames in the event more than one filename is assigned the same PID. Using PID portion 55 alone, the probability of the same PID being assigned to more than one filename is about 1 in  $2^{13}$  in one embodiment of the invention. Using PID portion 55 together with ~~DID~~ PIF 56 results in a probability of about 1 in  $2^{45}$  that the same PID will be assigned to more than one filename. In embodiments wherein all three portions, PID 55, ~~DID~~ PIF 56, and MCI 57 are utilized, the probability of the same PID being assigned to more than one filename is about 1 in  $2^{61}$  (virtually zero). In one embodiment of the invention, for each filename at its input, ~~Transform~~ transform 54 provides an MCI comprising 16 bits, a ~~DID~~ PIF comprising 32 bits, and a PID comprising 13 bits. It is to be understood that the terms, ~~DID~~ PIF and MCI are used herein merely for convenience in designating portions of the output of transform 54. Therefore, many designations for these bit portions could be devised and all such designations remain within the scope of the invention.

Please amend the paragraph beginning on Pg. 11, Line 1 of the specification as follows:

DC2 PACKETS

DC2 PACKETS. File manager 49 provides file contents 65 to packetizer 52. Packetizer 52 portions the contents of the file into at least one data packet. Packetizing allows for compressed video and audio images to be transmitted over high bandwidth channels such as satellite transmission. In embodiments of the invention wherein an MCI and a ~~DID~~ PIF are utilized, ~~Packetizer~~ packetizer 52 inserts the MCI and the ~~DID~~ PIF provided by transform 54 into at least one data packet along with file contents 65. In a data carousel embodiment of the invention, packetizer 52 provides data packets to carousel system 50. Carousel system 50 comprises a plurality of individual carousels 51, one carousel per PID. Packetizer 52 utilizes the PID ~~output~~ 55 of transform 54 to direct the data packets to a corresponding PID carousel 51 of data carousel system 50. Each PID carousel 51 comprises a plurality of packets corresponding to a file to be transported on that PID. In some embodiments thousands of PIDs comprise a carousel system.

Please amend Pg. 11, Line 22 – Pg. 12, Line 16 of the specification as follows:

FIG. 7 is a diagram of a DC2 packet 700 adapted for use in the invention. Packet 700 includes header 702, including a length field 706, an MCI 57, such as MCA 708, and other header information 704 used in the DC2 format. The user fields 710 include ~~the DID 56~~ payload identifier (PIF) 714, payload data 716 and other user defined fields 712. A cyclic redundancy code 720 follows the user fields 710.

Therefore, in embodiments of the invention employing an MCI and a ~~DID~~ PIF, each DC2 Packet comprises an MCI, a ~~DID~~ PIF and at least a portion of the data representing the contents of a given file. In many cases a plurality of DC2 packets will be utilized to convey the contents of a single file. Each of the packets corresponding to a given file is assigned the same PID as determined by transform 54.

MPEG2 PACKETS. ~~MPEG2 PACKETS.~~ Returning now to FIG. 3, multiplexer 61 combines inputs provided by carousel system 50, as well as audio 67, video 68, and other 69 packetized inputs to provide a data stream comprising

MPEG2 packets for modulation onto a six megahertz channel of the transmission medium 24. An example of an MPEG2 packet according to an embodiment of the invention is illustrated in ~~FIG. 5~~ FIG. 10. Each MPEG2 packet 100 comprises a header portion 110 and a payload portion 150. Together, header portion 110 and payload portion 150 comprise 188 bytes. Header 110 is further portioned into fields. Of particular interest in the present invention is PID field 112. This field comprises the PID corresponding to a file to be transported. Payload portion 150 comprises at least one DC2 packet. Typically a plurality of MPEG2 packets will comprise a single DC2 packet. However, according to an embodiment of the invention, all DC2 packets corresponding to a given file will be carried by MPEG2 packets having the same PID.

Please amend Pg. 12, Line 24 – Pg. 16, Line 29 of the specification as follows (note that this amendment amends a portion of the specification amended by Applicants in Applicants' Response, dated May 13, 2008, to the Office Action dated February 28, 2008):

DETAILED DESCRIPTION OF TRANSFORM. Turning now to FIG. 8, there is shown a functional block diagram of a transform 54 according to an embodiment of the invention. Transform 54 comprises a calculating function 300 and a PID look-up table (PLT) 305. First, the PLT 305 will be discussed.

PLT 305 is derived from the Program Specific Information (PSI) for an MPEG2 compliant system.

In order to construct the PLT of the invention, at least one range of PIDs available for transport of files is determined. In an MPEG implementation of the invention, PIDS are assigned in accordance with the requirements of the MPEG specification and the requirements of the specific system in which the invention is implemented from the ~~Program Specific Information (PSI)~~ PSI. PLT 305 is constructed by identifying available PIDs, listing the PIDs in a table (illustratively, PLT 305), and providing an index 307 into the listing 306. PLT 305 is transported from headend to receiver on a private data stream. In accordance with MPEG



convention, the PID for the private data stream used to transport PLT 305 is identified in the Program Map Table (PMT).

Calculating function 300 operates on respective filenames at its input to provide corresponding indices 307 at its output. Any calculating function having the following constraints is suitable for use in the invention. First, calculating function 300 provides a substantially unique, and repeatable index for each respective filename. Further, the indices provided by calculating function 300 have substantially a one to one correspondence to entries on the PLT 305.

In accordance with one embodiment of the invention, indices are calculated as follows. Respective file identifiers in the form of ASCII strings are provided to a checksum generator (CG) 320, for generating a Data Identifier (DID). In one embodiment of the invention, CG 320 performs a CRC Cyclic Redundancy Check (CRC) on the respective ASCII strings to provide a corresponding 64 bit checksum for each file. In alternative embodiments of the invention, a hash function is utilized to provide the ~~64-bit checksum~~ 64-bit DID. In other embodiments, the function is a pseudorandom number generator is utilized to provide the ~~64-bit checksum~~ 64-bit DID.

A selected 16 bit portion (designated X) of each ~~64-bit checksum~~ 64-bit DID is provided to a modulo divider 321. In one embodiment of the invention, the 16 MSBs Most Significant Bits (MSBs) of the ~~64-bit checksum~~ 64-bit DID comprise X. Modulo divider 321 provides a PID index ~~304~~ 307 at its output for each X at its input. The size of the PID index ~~304~~ 307 is chosen such that all of the PID entries available for use in PLT 305 can be accessed. The number of available PID entries listed on PLT 305 is denoted NPIDSON. In one embodiment of the invention, the PLT table 305 has 4096 entries and the index comprises 12 bits. Other embodiments of the invention have alternative sizes in accordance with the requirements of the system in which the invention is implemented. Modulo divider 321 operates on X to generate PID index ~~304~~ 307 in accordance with the relationship:

$$\text{PID index} = X \text{ modulo NPIDSON} \quad (1)$$

The PID index thus generated provides an index into the PLT 305.

As discussed above, some embodiments of the invention employ at least a second identifier, in addition to the PID, in the event that two filenames are assigned the same PID. In one embodiment of the invention, a ~~DID~~ PIF and an MCI comprise the remaining 32 and 16 Least Significant Bits (LSBs) respectively of the ~~64-bit output~~ 64-bit DID of CG 320. The choice of which bits, MSBs or LSBs, are utilized is a matter of convenience and either convention is suitable for use in the invention.

One of many alternative embodiments of a calculating function for providing a PID, ~~DID~~ PIF, and MCI is illustrated in FIG. 9. Filename FN is ~~provide~~ provided to CG 320. CG 320 provides a ~~64-bit checksum~~ 64-bit DID at its output. The ~~64-bit checksum~~ 64-bit DID is portioned into 4 consecutive ~~16-bit~~ 16-bit fields. The four ~~16-bit~~ 16-bit fields are provided to an XOR function 322 where they are combined in accordance with an exclusive ~~or~~ OR operation to provide a ~~16-bit~~ 16-bit output X to a modulo divider 321. As described above, modulo divider 321 operates on X to generate PID index ~~304~~ 307 in accordance with the relationship:

$$\text{PID index} = X \text{ modulo NPIDSON} \quad (1)$$

In similar fashion, two of the 16 bit fields are provided to a second XOR function 325. The 16 bit output of the second XOR function 325 is used as the MCI in embodiments of the invention which employ an MCI to lower the probability of collision.

In some embodiments of the invention, a ~~DID~~ PIF is formed by concatenating, i.e., stringing together, two ~~16-bit~~ 16-bit fields of the ~~64-bit output~~ 64-bit DID of CG 320. The concatenating operation is performed by concatenator (CAT) 330. In the embodiment illustrated in FIG. 9, the two least significant ~~16-bit~~ 16-bit fields are provided to concatenator 330. However, the choice of which two ~~16-bit~~ 16-bit fields are provided to concatenator 330 is not critical to the invention. Any two ~~16-bit~~ 16-bit fields can be utilized. However, the probability of the event that two filenames are assigned the same PID is related to the degree of correlation between the PID, the MCI and the ~~DID~~ PIF. The lower the correlation, the lower the probability of that event. Therefore, in

embodiments of the invention wherein the lowest correlation is desired, the two ~~46-bit~~ 16-bit fields chosen to form the DID are different from the two ~~46-bit~~ 16-bit fields chosen as a basis for the MCI.

Substantially identical transforms operate in both the sender and the receiver. Both the sender and the receiver employ substantially identical PLTs. The receiver acquires the PLT 305 when it first tunes to the in data band channel.

#### CHANGES AT THE FILE SERVER

The contents of file server 46 are subject to change. File events such as file removal, moving in the directory structure and file updates, to name a few, can occur. In many cases, these changes are relevant to at least one application program 66 running in receiver 28. In one embodiment of the invention, receiver 28, and application program 66 is periodically notified of these types of events. In one embodiment of the invention, notification of file events is provided to receiver 28 via an MPEG2 private data channel. The PID for the private data channel is provided in the PMT. Headend 22 constructs and delivers events and partial directories (referred to as "directory markers") as they change, eliminating the need for the file manager ~~206~~ 49 to maintain and permanently store them, and providing real-time synchronization of events and markers with the files to be transported. The available CRC information is also fed back to the server ~~202~~ 46, which uses the information for creating standard and alternate markers containing subsets of the directory information.

The exemplary golden PID, such as, for example, PID 600 depicted in FIG. 6 and discussed below in greater detail, carries two bitmaps and the starting PID number. The first bitmap is the PID allocation bitmap which indicates which PIDs are used in the PID calculation.

The second bitmap is the PID usage bitmap (described ~~above~~ below), which indicates which PIDs are currently in use. This is a subset of the allocated PIDs. The PID usage bitmap is a standard bitmap shipped with the least significant bit first, i.e., bit 7 of byte 0 corresponds to the first PID in the map.

"FILE NOT FOUND" AND COLLISION CONDITIONS CONDITIONS.

A receiver calculates the CRC and PID for a desired file, writes the PID to a PID filter, and waits for MPEG2 data packets assigned to that PID. If the desired data packets arrive within a predetermined period of time, then the file is received. If the predetermined period of time passes, and the receiver has not received the desired unit of data, a "file-not-found" condition exists in which the desired data are considered unavailable. In other embodiments, such as the examples described below, additional features are provided to more rapidly identify a file-not-found condition, to reduce the amount of time it takes the receiver to detect a file not found condition.

Reference is again made to ~~FIG. 2~~ FIG. 3. The file manager ~~206~~ 49 includes a directory mapper (described below with reference to FIG. 4) associated with the representative number; the directory mapper generates and transmits to the set top apparatus ~~250~~ box 28 a partial directory (e.g., marker 416 shown in FIG. 4) identifying each of the at least one file that has respective directory information that transforms to that same PID.

In some embodiments, to detect a file-not-found (FNF) condition with a reduced latency, a combination of strategies is used. Directory "markers" (as shown in Tables 1 and 2, below) are spun periodically on each PID that carries data, as described in greater detail with reference to FIGS. 4-6. The standard markers contain the 64-bit DID number of each file being spun on the PID over which the marker is sent. In these embodiments, the markers provide PID specific directory information. If the ~~settop-250~~ set top box 28 sees one of these markers before it sees the file it's attempting to load, it reads the marker to determine whether the file actually exists. The marker spin rate is determined empirically by balancing bandwidth consumption against the desired FNF detection time.

FIGS. 4 and 5 show the identifier calculation, collision avoidance and FNF latency reduction methods. FIG. 4 is a flow chart diagram showing one example of how a sender (e.g., a cable headend) performs the ~~DID~~ PIF and PID calculation and transmits the data.

Please amend the paragraph beginning on Pg. 17, Line 3 of the specification as follows:

At step 404, the sender performs the transform by generating a number with an approximately uniform probability distribution, as a function of a bit sequence (e.g., file name) associated with a unit of data (e.g., packet, file) to be transmitted. The term "function" implies that a given bit sequence only produces one value, in a reproducible manner. If the same bit sequence is input to the transform function three times, the same output is generated three times. In the exemplary embodiments, the generated number is the ~~64-bit binary number (DID)~~ 64-bit DID described above. The 64-bit length is selected to provide an extremely small likelihood that two different file names will yield the same calculated number. Some embodiments use smaller number lengths to provide any desired probability that two file names will not yield the same number. Other embodiments use larger number lengths to provide any desired probability that two file names will not yield the same number.

Please amend the paragraph beginning on Pg. 17, Line 24 of the specification as follows (note that this portion of the specification was previously amended by Applicants in Applicants' Response, dated May 13, 2008, to the Office Action dated February 28, 2008):

where PID index is an index associated with the PID; X is a result of performing at least one XOR operation on at least two portions of the DID (e.g., cyclic redundancy code, hash function, pseudorandom number); and NPIDSON is a number of allocated PIDs, which, in one embodiment, corresponds to a number of packet processors to which payload files are being sent.

Please amend Pg. 18, Lines 8 – 19 of the specification as follows:

At step 410, a second portion of the number is used as a multicast identifier. In some embodiments, the multicast identifier is formed by performing an XOR operation on two non-contiguous portions of the ~~PIF~~ DID. In the exemplary embodiment, the multicast identifier is a 16-bit number determined by XORing portions A and C of the 64-bit DID.

To allow the receiver to initially recognize a desired unit of data (e.g., packet or file) based on the multicast identifier, it is desirable for the system to assign a unique multicast identifier to each unit of data. However, because there ~~[[is]]~~ are multiple units of data transmitted with the same PID, and there are only 65,536 possible 16-bit multicast ID's, it is possible for two files with two different bit sequences (e.g., two different file names) to have the same multicast ID. Thus, in this embodiment, a multicast collision detection and correction feature is provided.

Please amend the paragraph beginning on Pg. 18, Line 30 of the specification as follows:

At step 416, in some embodiments, the sender periodically transmits (spins) a standard marker packet or message that includes a list of all of the 64-bit DIDs for the sets of at least one packet (e.g., packets, files, messages) to be sent using the given PID on which that marker is sent. In other embodiments, a file is sent instead of a marker. The marker is a partial directory that only covers the units of data spun on a single PID. A marker is spun on each PID carrying broadband data. In some embodiments, the marker spins at the rate of once every 5-10 seconds. Higher spin rates are possible, but consume more bandwidth. In the exemplary embodiment, the standard marker is a packet, as opposed to a file, and the calculation of ~~PID~~ DID MCI, PIF, and PID does not apply to the marker. Instead, a predetermined PID is used to send the markers.

Please amend the paragraph beginning on Pg. 20, Line 8 of the specification as follows:

For a relatively large multicast address (e.g., 16-bits) and relatively large number of PIDs (e.g., 2000) relative to the number of units of data (e.g., 35,000), there is a low probability of two different sets of at least one packet having the same multicast ID. In other words, in the vast majority of cases, for any given PID, any two units of data having the same multicast ID have a high probability of corresponding to the same set of at least one packet. Thus, packet recognition by a recipient is generally achieved more rapidly by comparing the calculated multicast ID to the multicast ID of a received unit of data. That is, packets having

a different multicast address from that of the desired file [[is]] are quickly eliminated from consideration without examining the PIF (32 MSBs of the 64-bit DID) contained in the packet. The PIF is only examined if the desired multicast address is found. However, in preferred embodiments, a multicast collision detection and correction mechanism is desired to positively identify when a multicast collision exists, and eliminate the condition.

Please amend Table 2 (beginning on Pg. 21, Line 6 of the specification) as follows

Table 2

$b_{1,1}$ to $b_{1,16}$	$b_{1,17}$ to $b_{1,32}$	$b_{1,33}$ to $b_{1,48}$	$b_{1,49}$ to $b_{1,64}$	<u>[[MC<sub>1</sub>]]</u> <u>MCA<sub>1</sub></u>
$b_{2,1}$ to $b_{2,16}$	$b_{2,17}$ to $b_{2,32}$	$b_{2,33}$ to $b_{2,48}$	$b_{2,49}$ to $b_{2,64}$	<u>[[MC<sub>2</sub>]]</u> <u>MCA<sub>2</sub></u>
$b_{3,1}$ to $b_{3,16}$	$b_{3,17}$ to $b_{3,32}$	$b_{3,33}$ to $b_{3,48}$	$b_{3,49}$ to $b_{3,64}$	<u>[[MC<sub>3</sub>]]</u> <u>MCA<sub>3</sub></u>
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
$b_{n,1}$ to $b_{n,16}$	$b_{n,17}$ to $b_{n,32}$	$b_{n,33}$ to $b_{n,48}$	$b_{n,49}$ to $b_{n,64}$	<u>[[MC<sub>n</sub>]]</u> <u>MCA<sub>n</sub></u>

Please amend the paragraph beginning on Pg. 22, Line 13 of the specification as follows:

At step 420, the sender periodically transmits a PID usage map.

(Although FIG. 4 shows the PID usage map being spun after the standard and alternate markers, the markers are spun on each PID after the PID usage map is spun.) The PID usage map identifies which PIDs are being used to transmit data. For example, in an exemplary system ~~wherein~~ in which 13 bits are used to specify the PID, and about 2000 PIDs are actually used, about one fourth of the 8192 available PIDs are used. The PID usage map identifies which of the PIDs are used. In some embodiments, the PID usage map is a packet (PID usage bitmap) having one bit for each PID, with the value of each bit indicating whether

that PID is being used. In some embodiments, the PID usage map is sent using a separate PID that is shared by all of the receivers referred to herein as the "~~golden PID.~~" "golden PID". In some embodiments, the PID usage map is sent in a message instead of an alternate marker. In other embodiments, the PID usage map is sent in a file instead of a marker.

Please amend the paragraph beginning on Pg. 23, Line 16 of the specification as follows:

At step 502, the receiver calculates a DID, PIF, ~~PID~~ and a multicast-ID PID, and MCI from the bit sequence (e.g., file name) associated with the desired unit of data, using the same calculation that is used by the sender (as described above with reference to FIG. 4). Mapping of the 64-bit DID to a PID is done using a PID usage map and a modulus (PID index). The settop uses the PID usage map acquired when it first tunes to the inband channel. The PID usage map provides the settop with a starting PID number (which is the first PID in the PID usage map), followed by a Run-Length-Encoded (RLE) bitmap whose "on" bits indicate a PID in use. The modulus is performed on the number of "on" bits in the map. Once the PID index is determined, a number of on-bits in the PID allocation map are counted, and the PID offset corresponds to the number of counted PIDs. Then the PID number is computed by adding the start PID to the actual PID offset of the "on" bit within the map.

Please amend Pg. 25, Line 4 – Pg. 26, Line 21 of the specification as follows

At step 525, the alternate marker is checked, to determine whether the DID corresponding to the desired unit of data is listed in the alternate marker. If that DID is not listed in the alternate marker, ~~then~~ thenm at step 526, a file-not-found condition is detected.

If the DID corresponding to the desired unit of data is listed in the alternate marker, then at step 527, the receiver reads the non-colliding multicast-ID MCI from the alternate marker.

At step 528, the receiver waits for the unit of data with the non-colliding multicast-ID MCI.



At step 530, when the packet having the non-colliding ~~multicast-ID~~ MCI is received, then the desired unit of data is found.

Variations of the method of FIG. 5 are contemplated. For example, in some embodiments, instead of using the 32-bit PIF to make the final comparison, the full 64-bit DID calculated by the server is included in each packet, and the receiver compares the full 64-bit DID of the received packet to the number contained in the marker. This provides an additional 16 bits that are independent of the ~~multicast-ID~~ MCI (as well as 16 bits that overlap with those used to form the ~~multicast-ID~~ MCI), for a total of 64 bits of entropy. The process flow is similar to that described above, except that at steps 514 and 520, the comparison is made to the 64-bit DID, instead of using the 32-bit PIF. In other embodiments, the PIF is composed of a different portion of the 64-bit DID than described above. In one example, the multicast address is formed by XORing segments A and C of the 64-bit DID, and the PIF is formed by concatenating segments B and D.

Overhead associated with the exemplary system described above is estimated as follows: Each file piece will increase by 10 bytes (6 additional bytes for the ID and 4 for the file length). A directory marker consuming one MPEG packet (188 bytes) every 10 seconds on every PID results in 700 packets per second in a 7000 PID map (Maximum packet rate is 17952/second). One DC2 packet will suffice for the PID map, which is cached. An event consumes 10 bytes in raw data; so about 16 events comprise a single MPEG packet. This works out to 5-6% of the bandwidth consumed by an exemplary file manager 206 49.

Given a file name, the exemplary "inband directory" system is able to provide fast lookups for existing files, as well as reasonably fast detection of non-existent files. It provides timely notification of changes to files (version changes) in a way that is easily monitored by the set top 250 box 28. Synchronization between the directory (standard and alternate markers) and the files it represents are implicit. Minimal inband and settop resources are consumed. Inband bandwidth consumption is low, and settop CPU loading and memory usage is kept to a minimum. Lookup latency is optimized for speed, and the latency

associated with lookup failure (files not found) is reasonably short. Large numbers of files are supported. Although 10,000 to 30,000 is considered nominal, the system is used to spin a larger number of files.

FIG. 6 is a diagram of data being transmitted via several PIDs. PIDs bearing no data carry no markers. Instead, a "PID usage" bitmap is sent on a separate PID 600 (referred to as the "golden PID") having a predetermined PID number. Once the receiver 250 28 determines the PID of the desired file, the receiver 250 28 checks the PID usage bitmap to determine whether that PID is currently in use. If the PID derivation calculation lands on an unused PID (bit is "off"), then a ~~FNF~~ file-not-found is returned.